LAB4 WRITEUP

Submission Questions:

18.

a) What operating system (including revision) did you use for your code development?

Ans: Windows 11

b) What compiler (including revision) did you use?

Ans: SDCC

c) What exactly (include name/revision if appropriate) did you use to build your code (what IDE, make/makefile, or command line)?

Ans:STM32CubeIDE 1.13.2 and CodeBlocks:SDCC

d) Did you install and use any other software tools to complete your lab assignment?

Ans No.

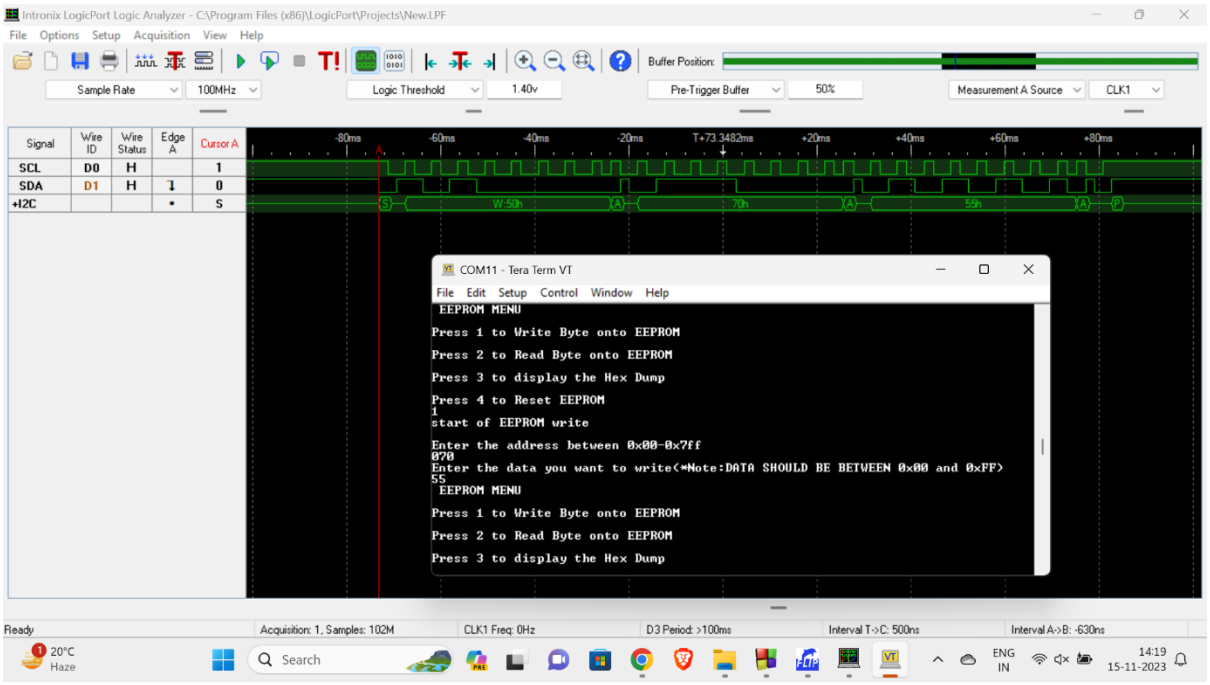
e) Did you experience any problems with any of the software tools? If so, describe the problems.

Ans: No

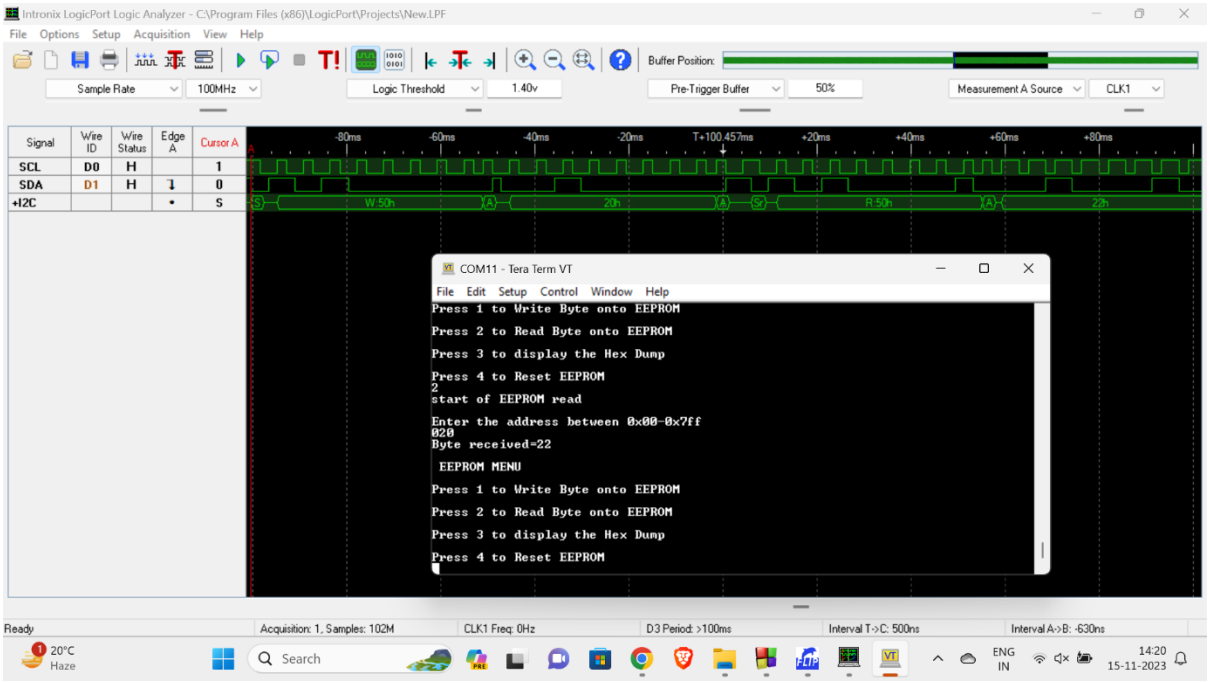
Pictures/Screenshots:

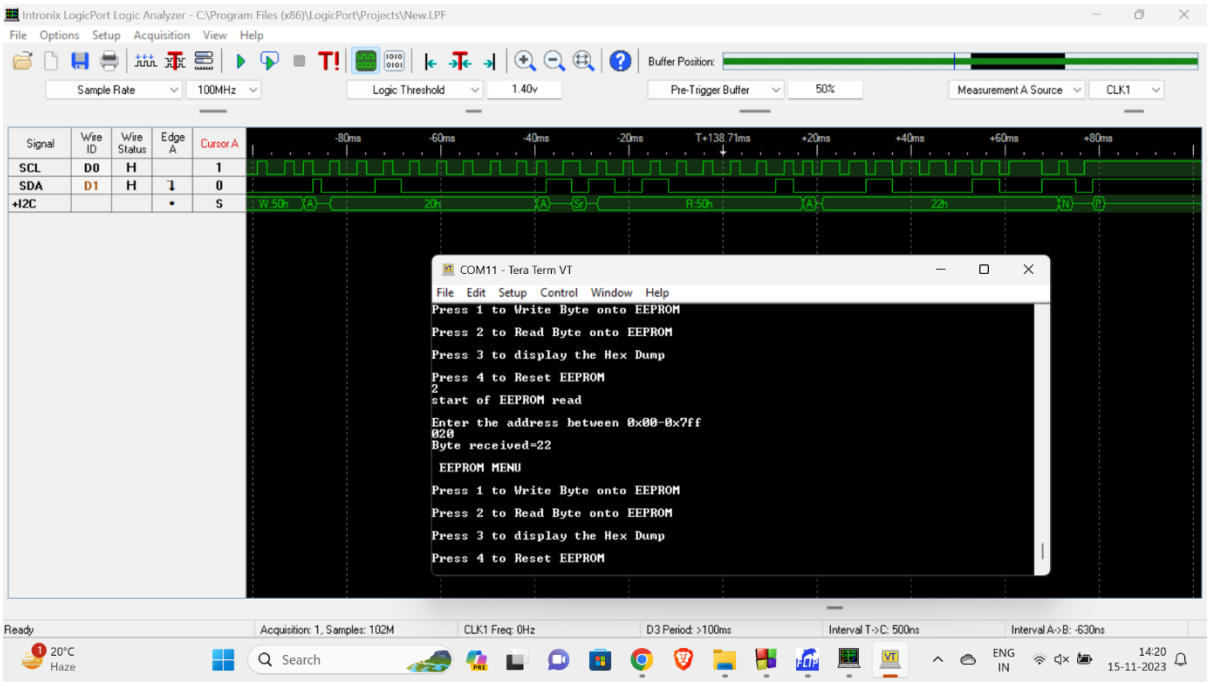
PART1: EEPROM

1. Screenshot of Terminal output and Logic analyzer timing diagrams during EEPROM write.

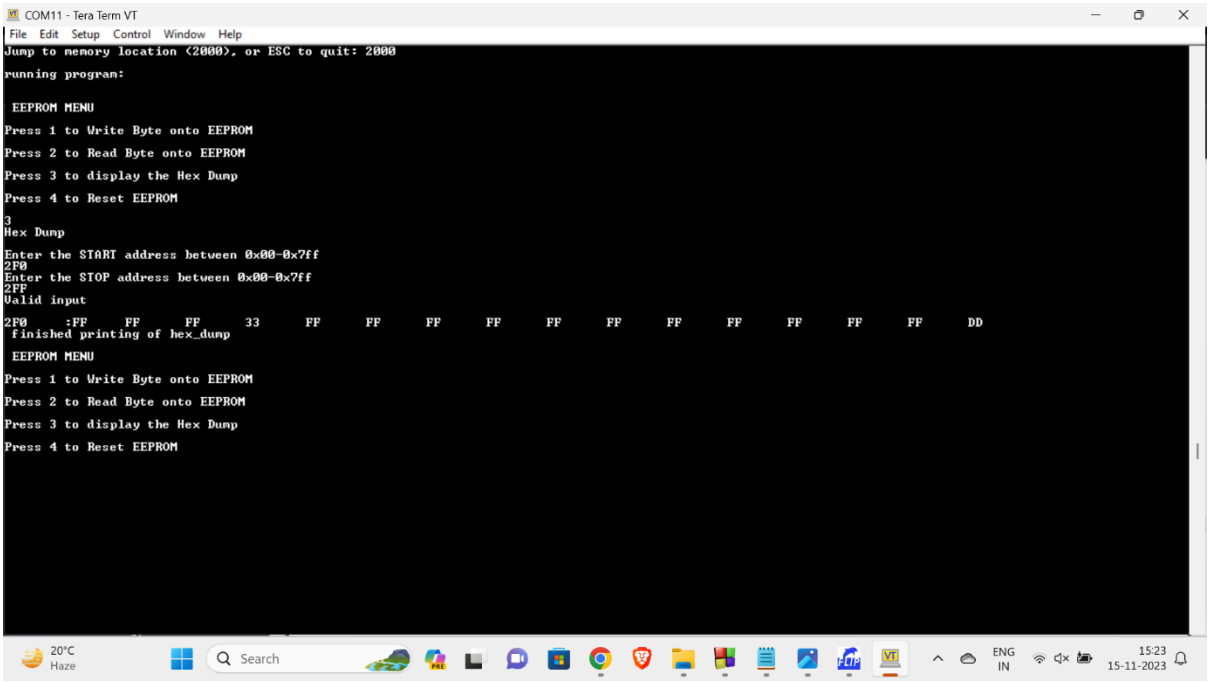


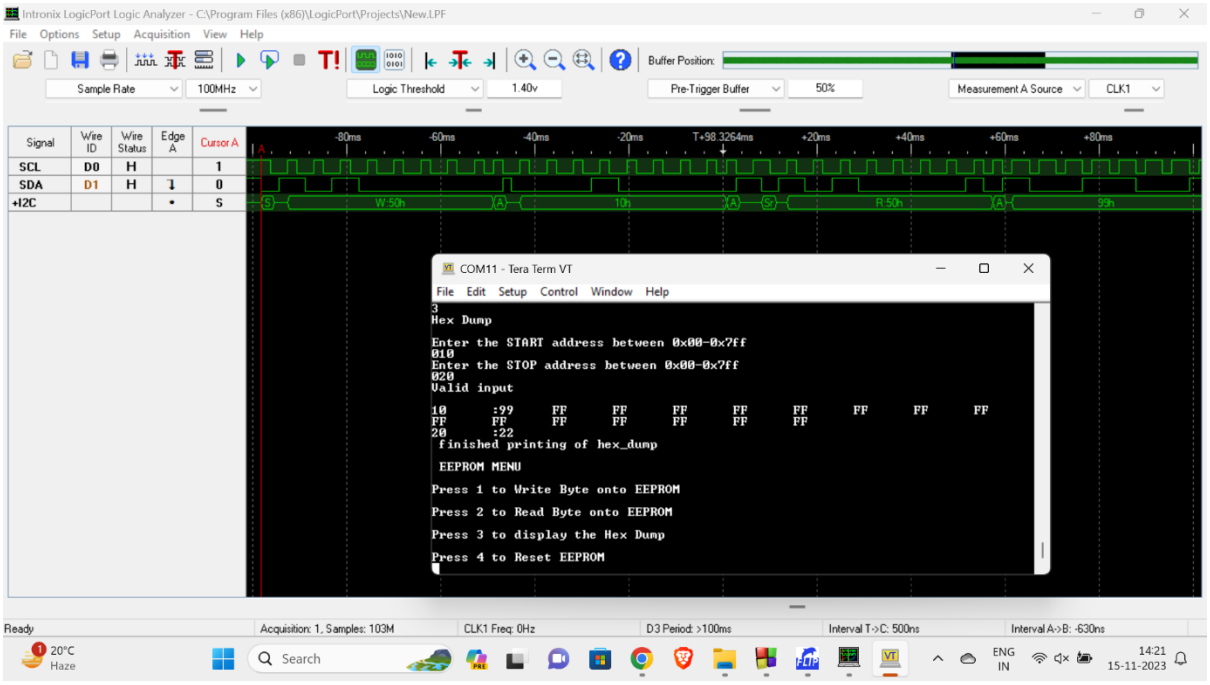
2. Screenshot of Terminal output and Logic analyzer timing diagrams during EEPROM read.



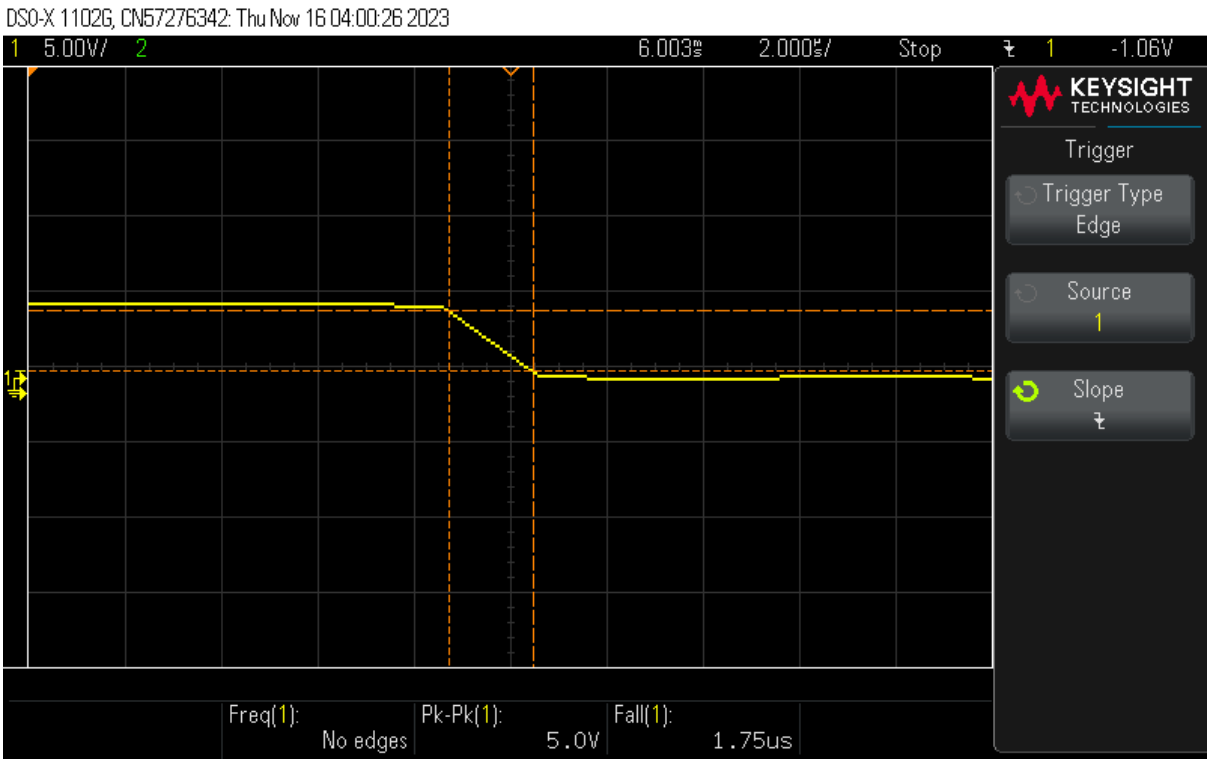


3. Screenshot of Terminal output and Logic analyzer timing diagrams during EEPROM Hex Dump.



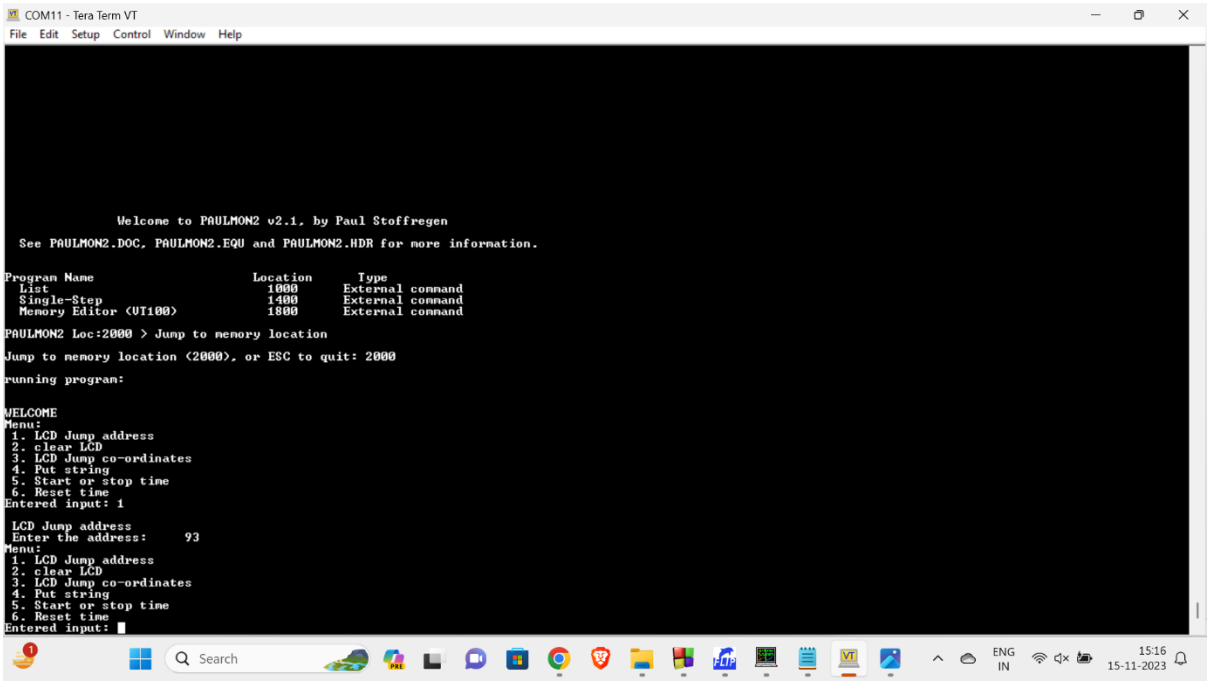


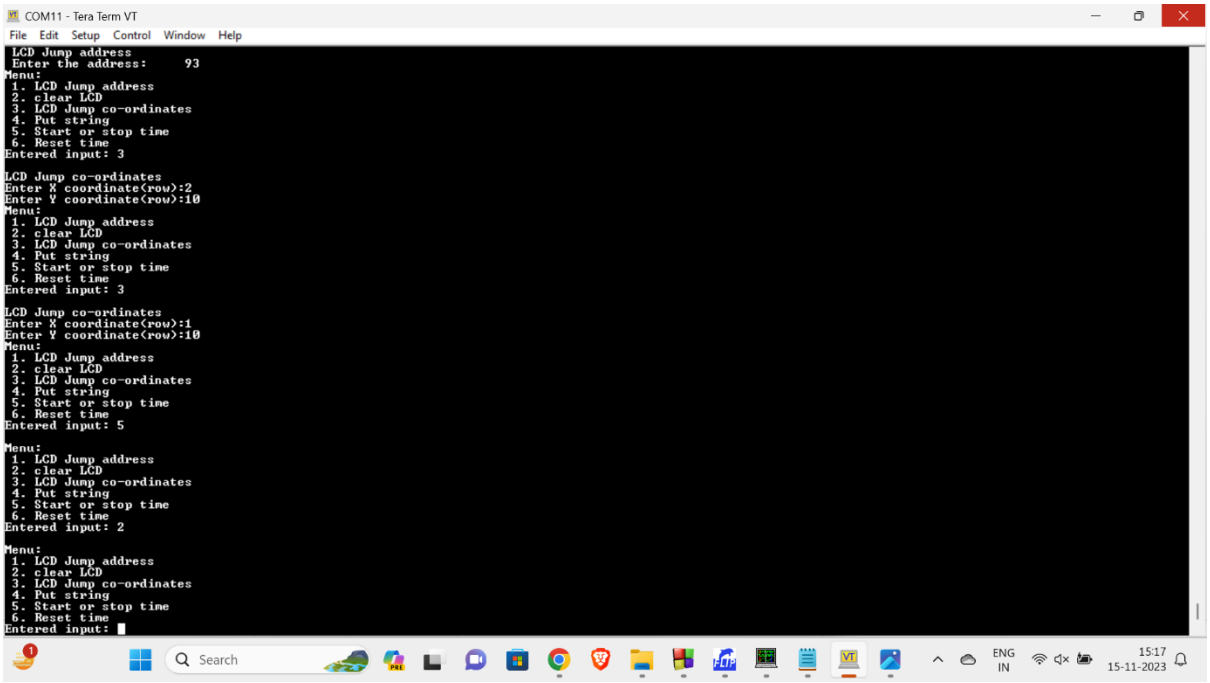
4.Oscilloscope picture of SDA falltime



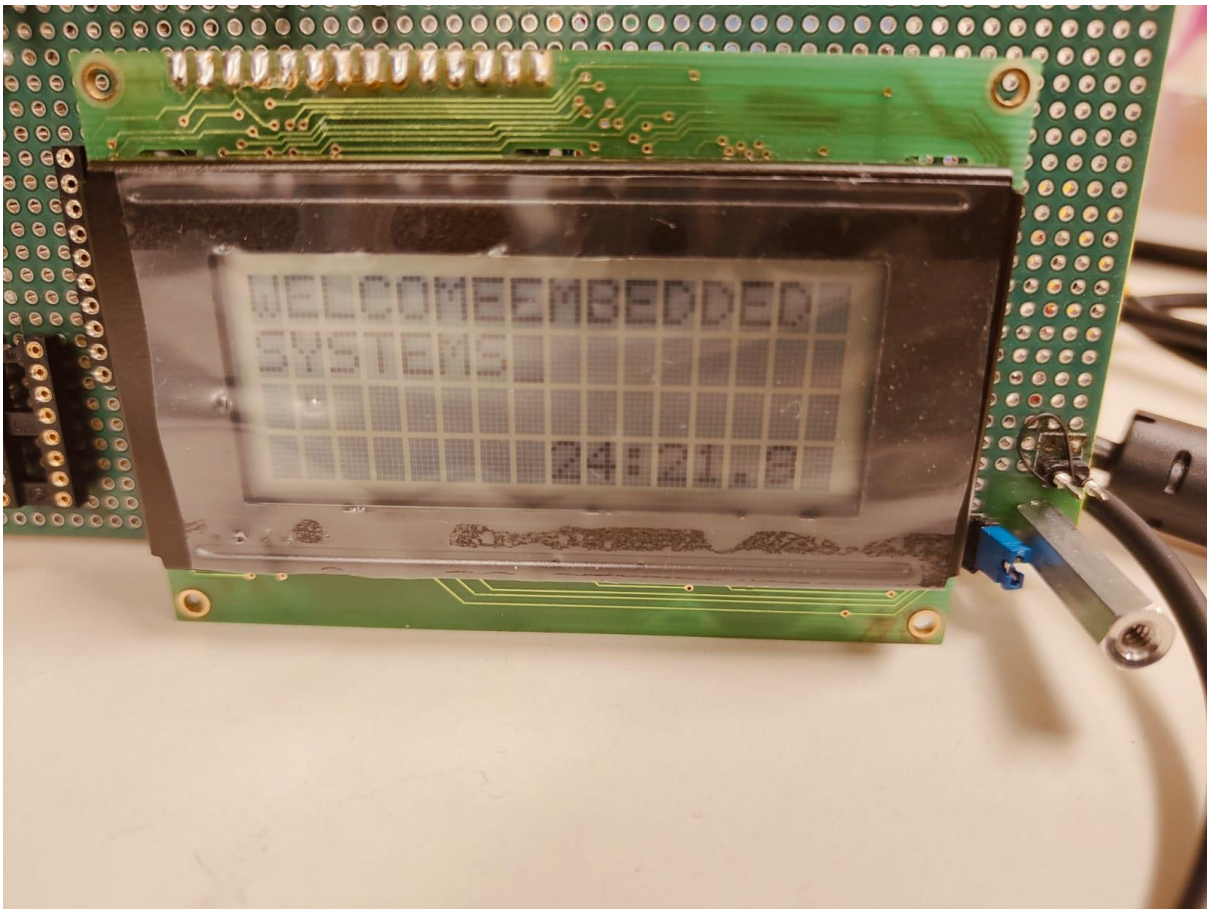
Part 2:

1.Screenshots of Terminal output of LCD.

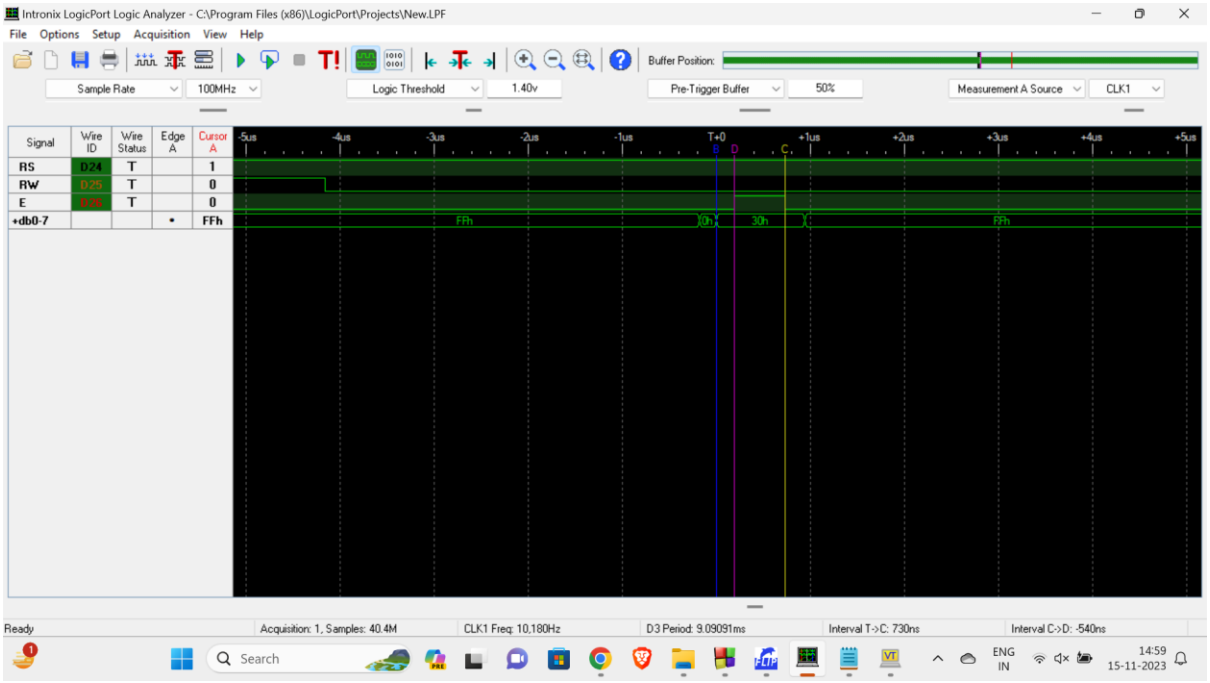




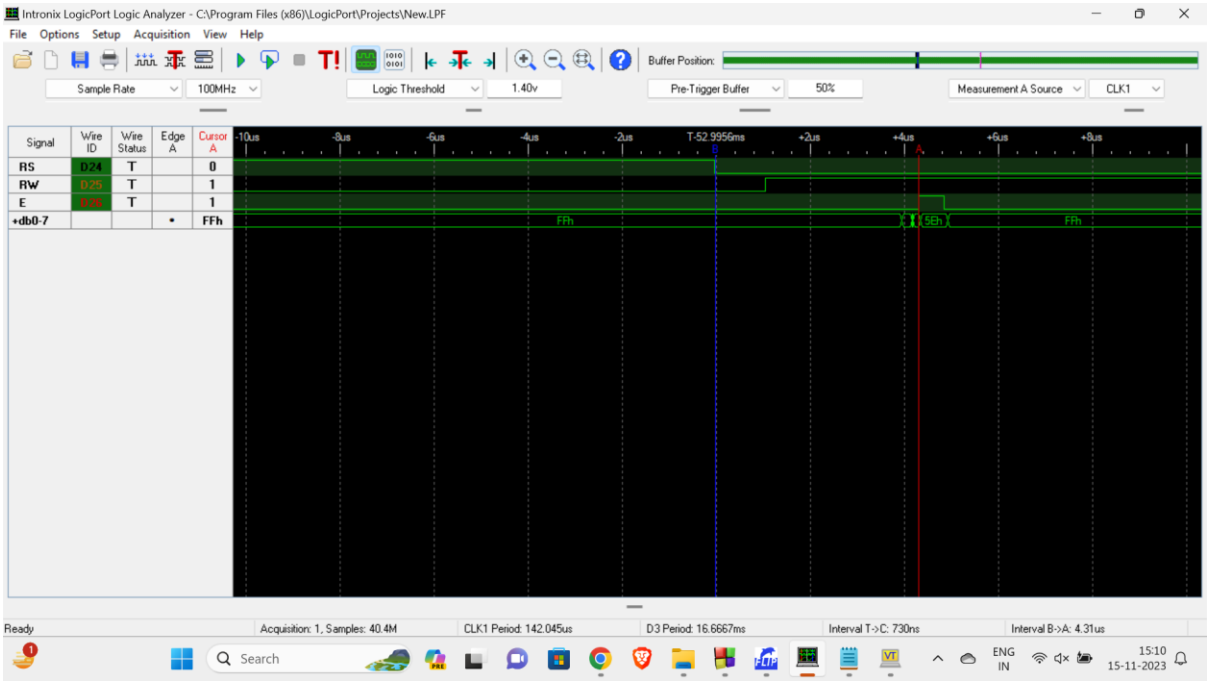
2.Screenshot of on-board LCD screen.



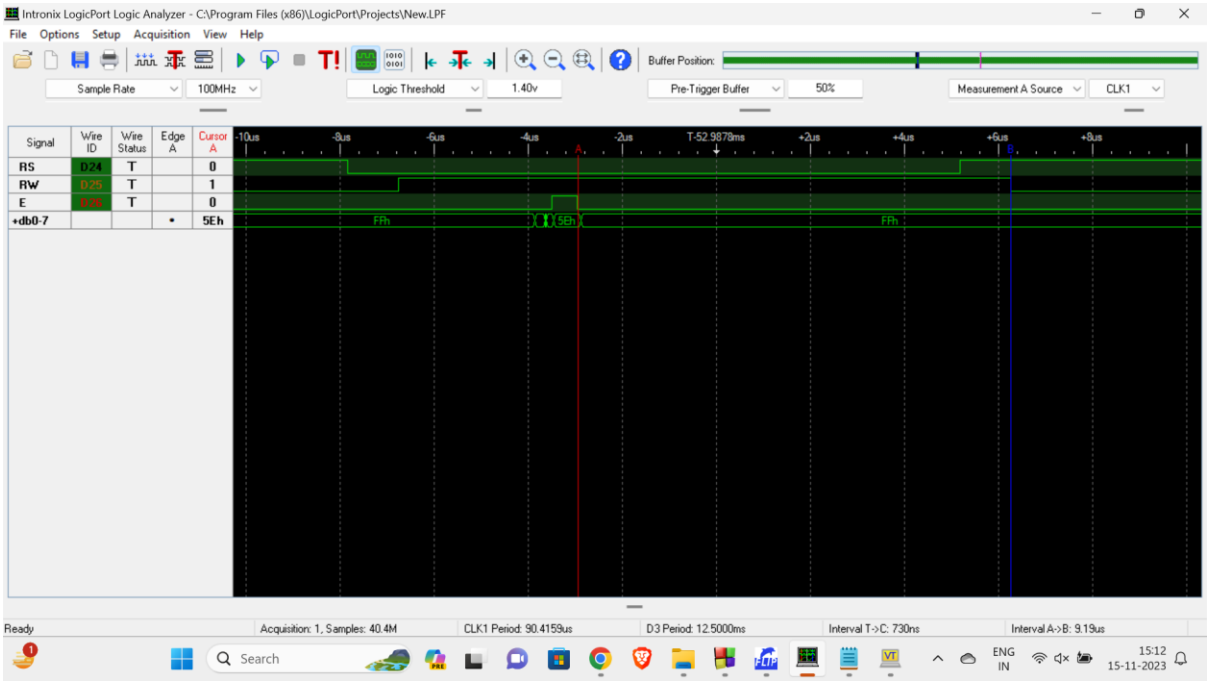
4. Logic analyser screenshot measuring the pulse width of Enable which should be a minimum of 230ns.
   Enable pulse width = 540ns (meets the requirements)



5. Logic analyser screenshot measuring the Address setup time which should be a minimum of 40ns.
   Address setup time = 4.31us (meets the requirements)

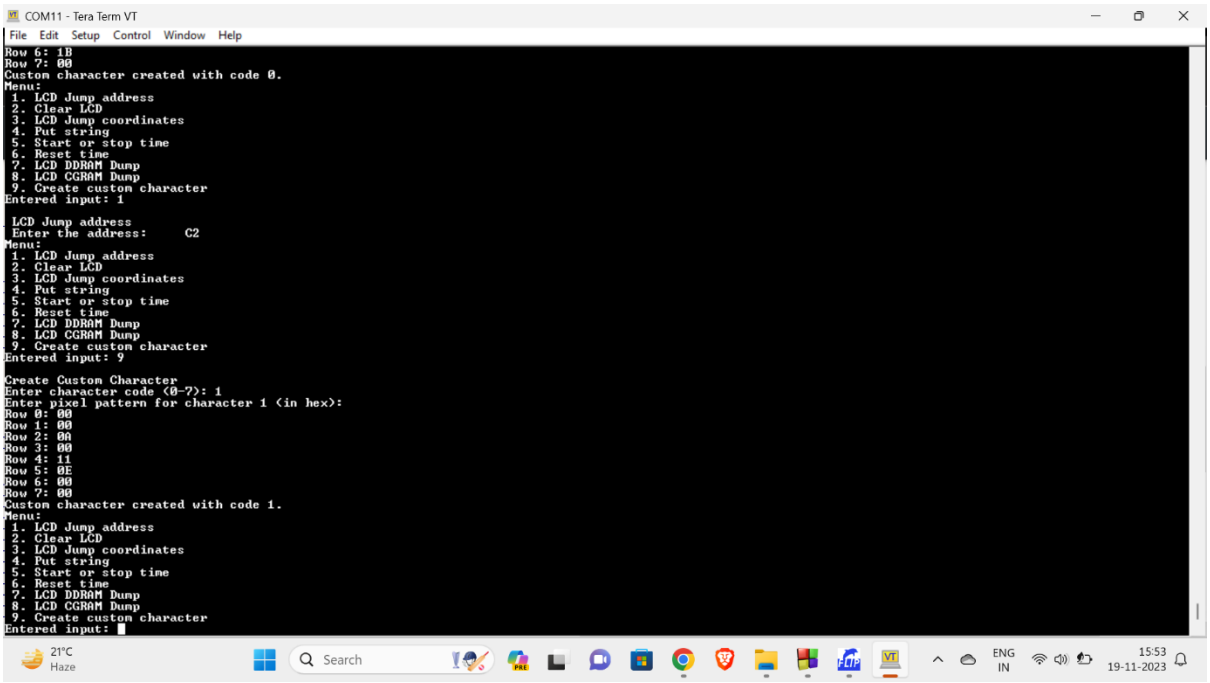


6. Logic analyser screenshot measuring the Address hold time which should be a minimum of 10ns.
   Address setup time = 9.19us (meets the requirements)

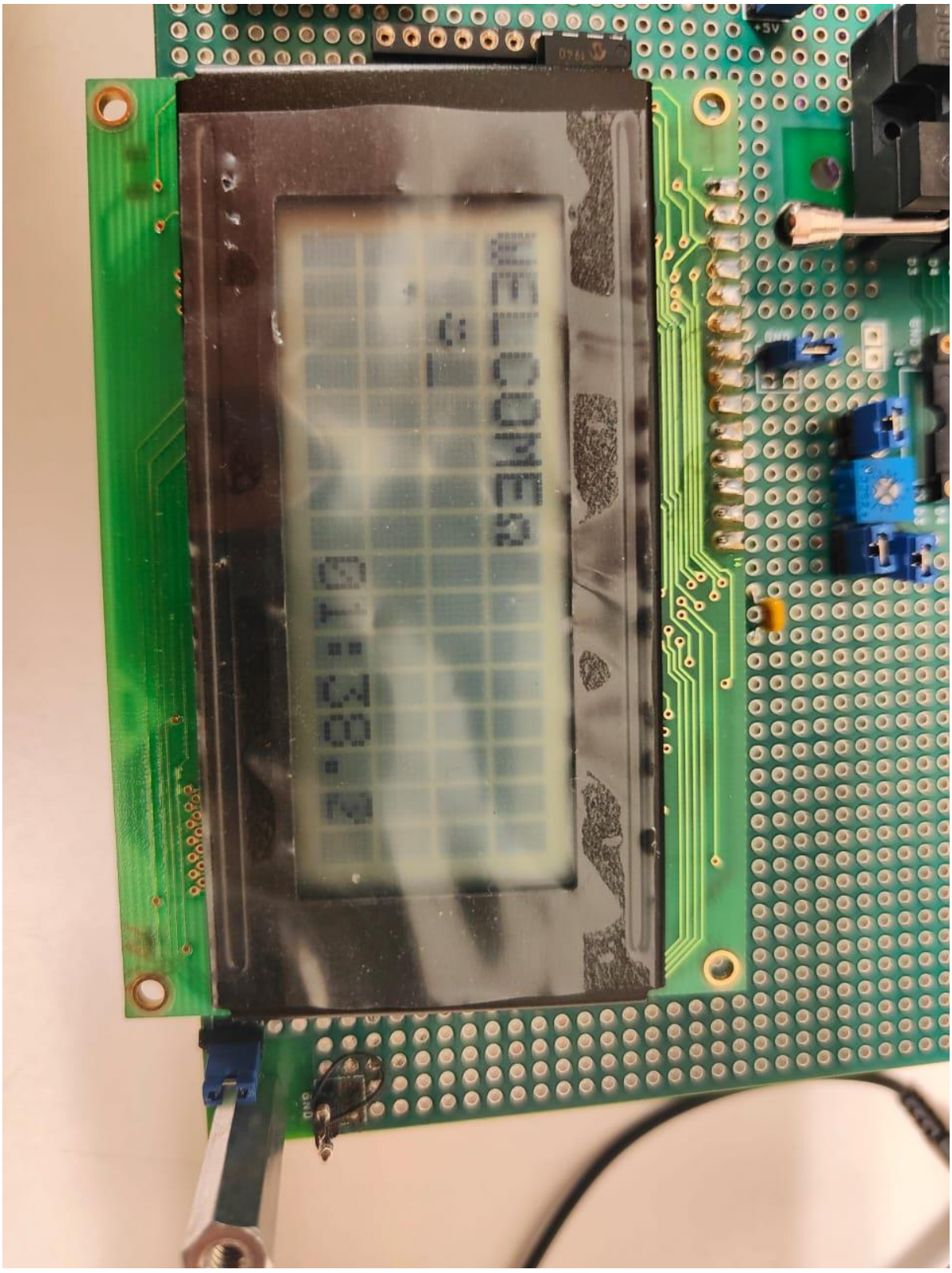


PART 3:

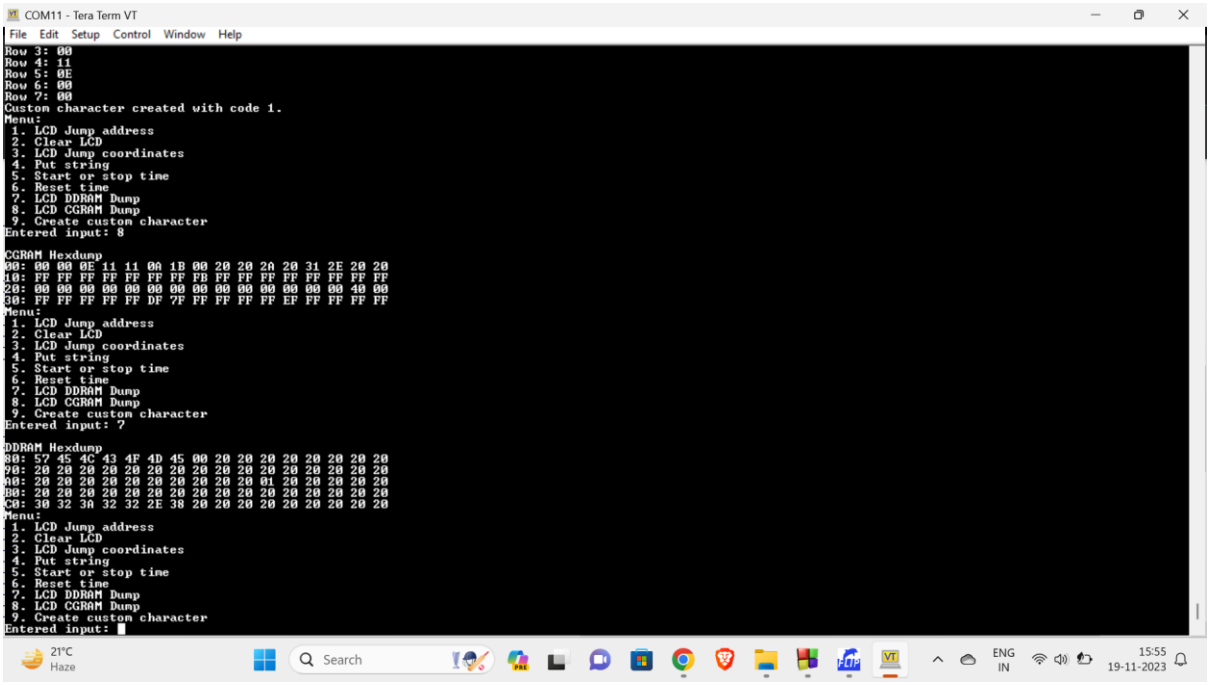
1. Screenshots of output terminal for creating a custom character.



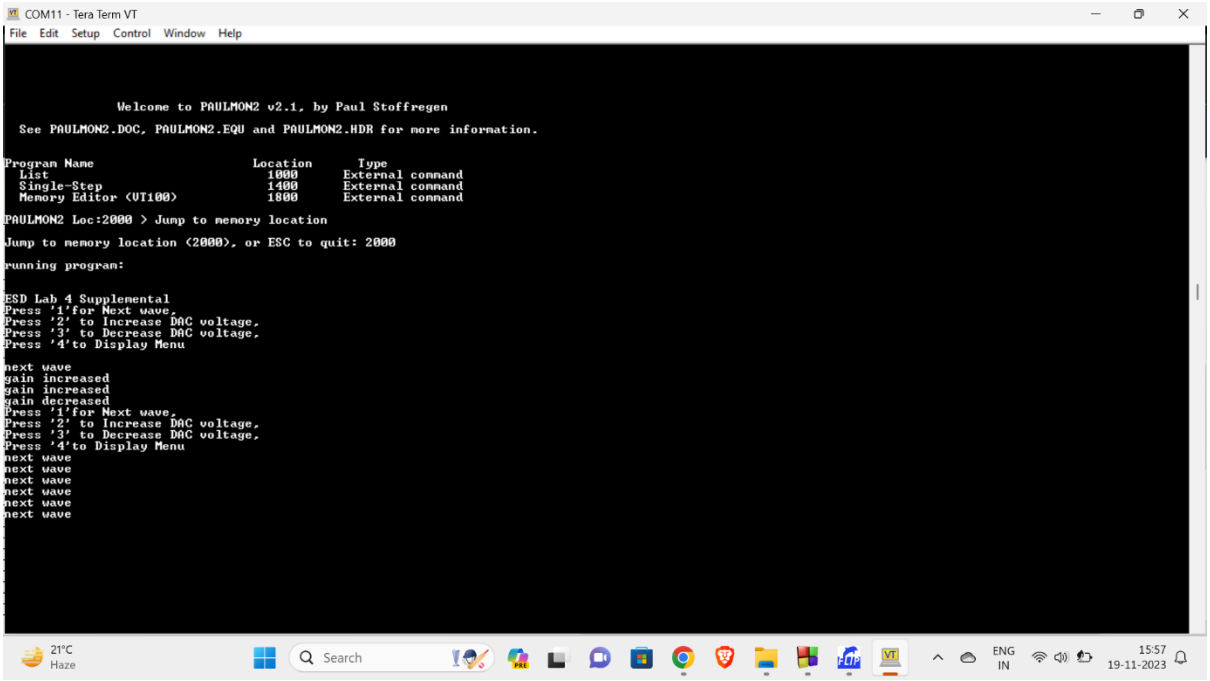
Screenshot of on-board LCD screen.



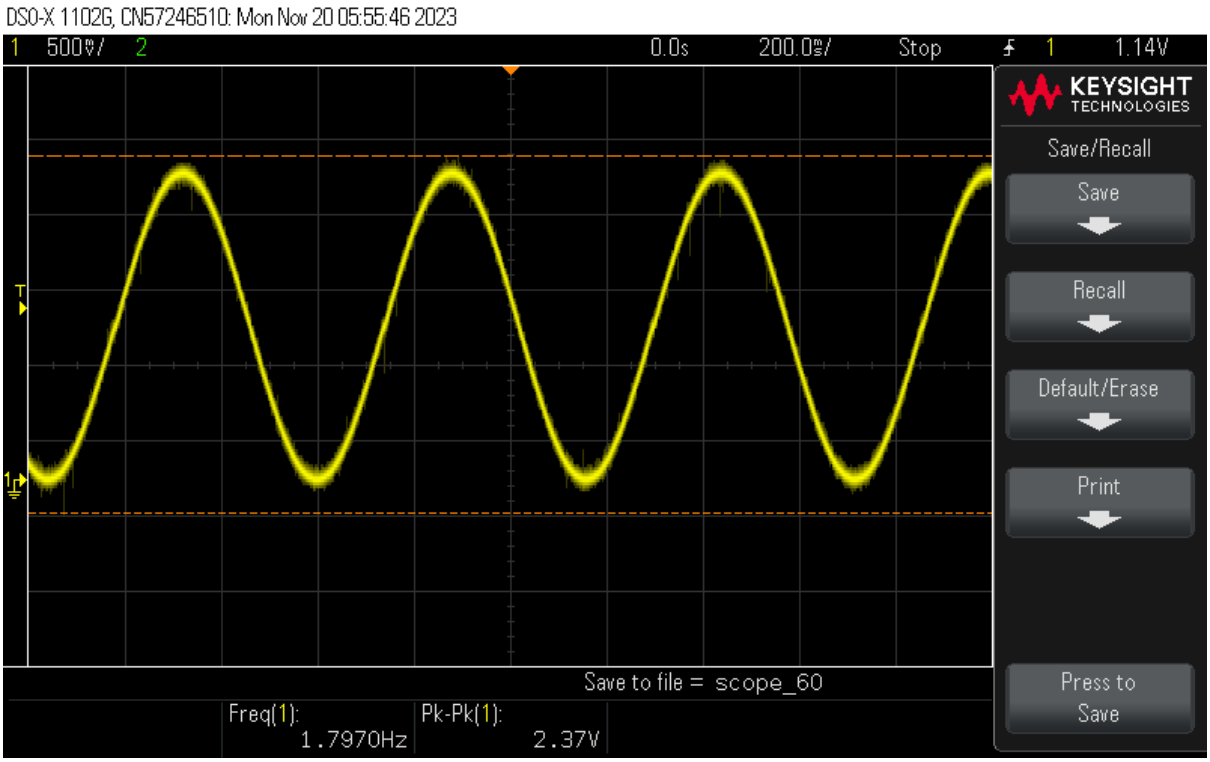
Screenshot of Terminal output with CGRAM Hexdump and DDRAM Hexdump.



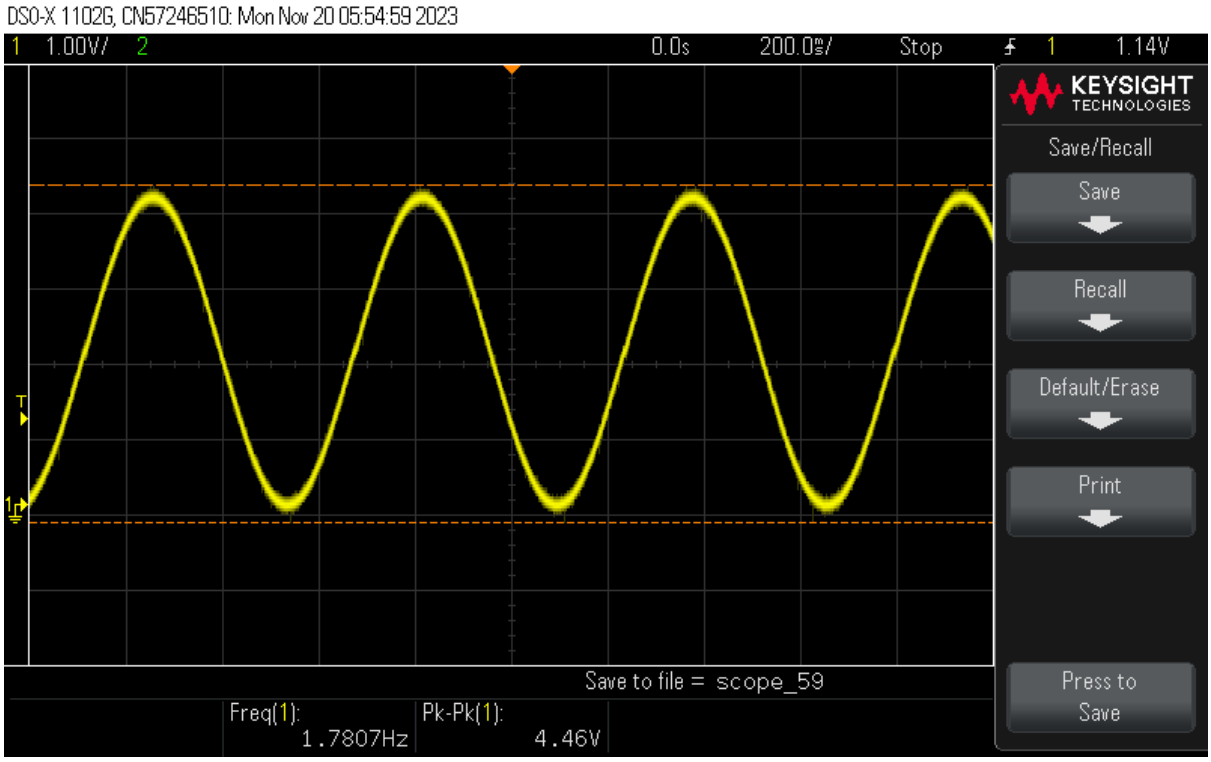
2. Screenshots of Terminal output for DAC implemented using SPI protocol.



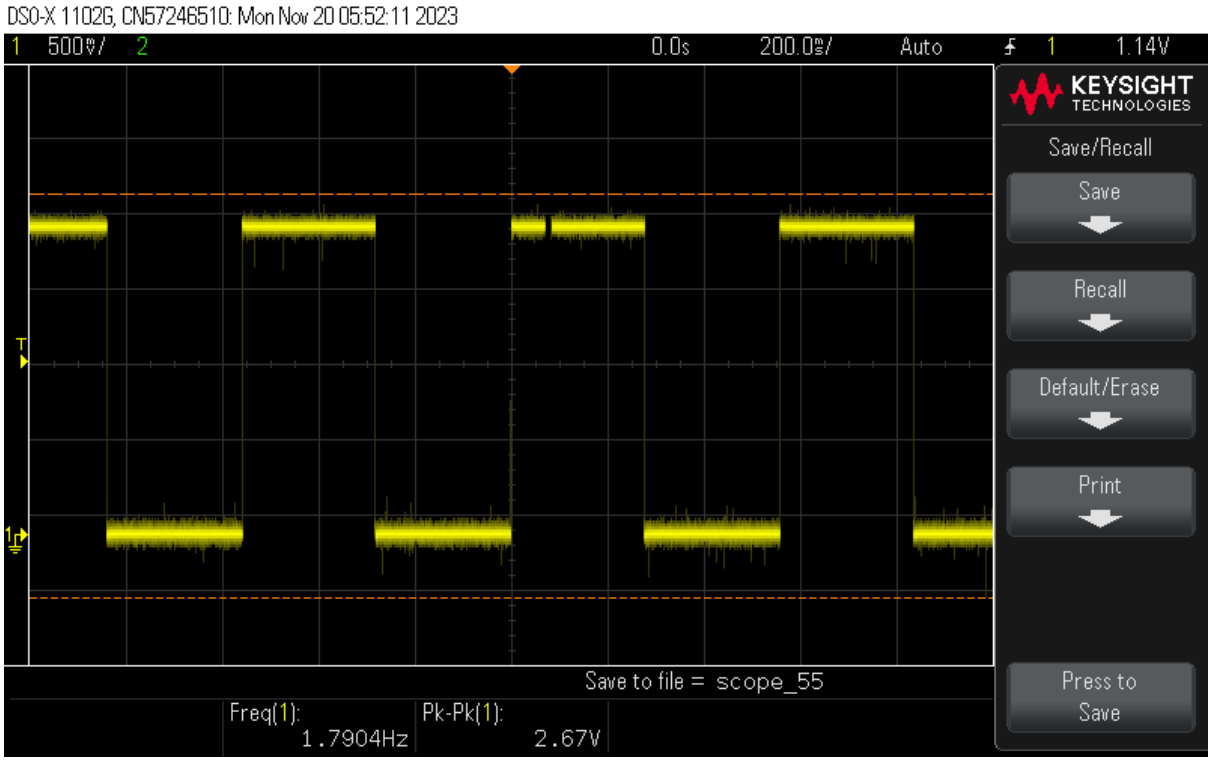
Oscilloscope picture of DAC used to generate Sinusoidal wave.



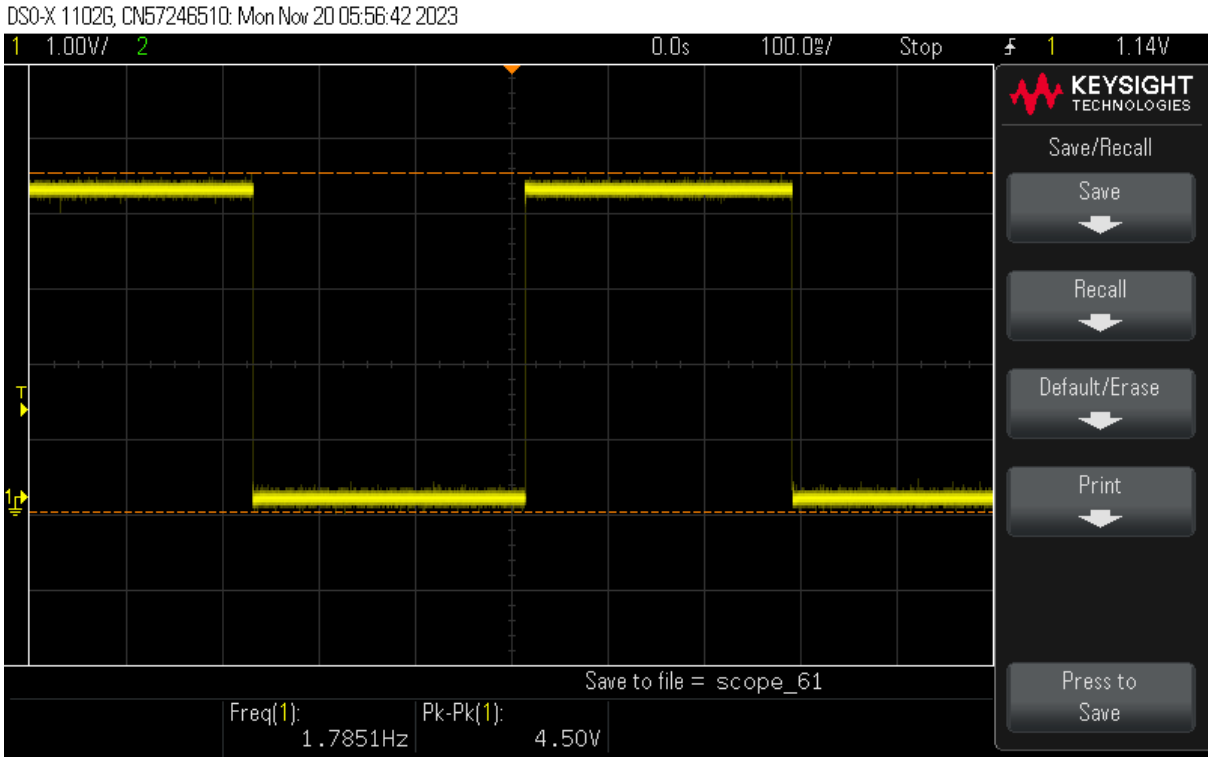
Oscilloscope picture of DAC used to generate Sinusoidal wave with increased gain.



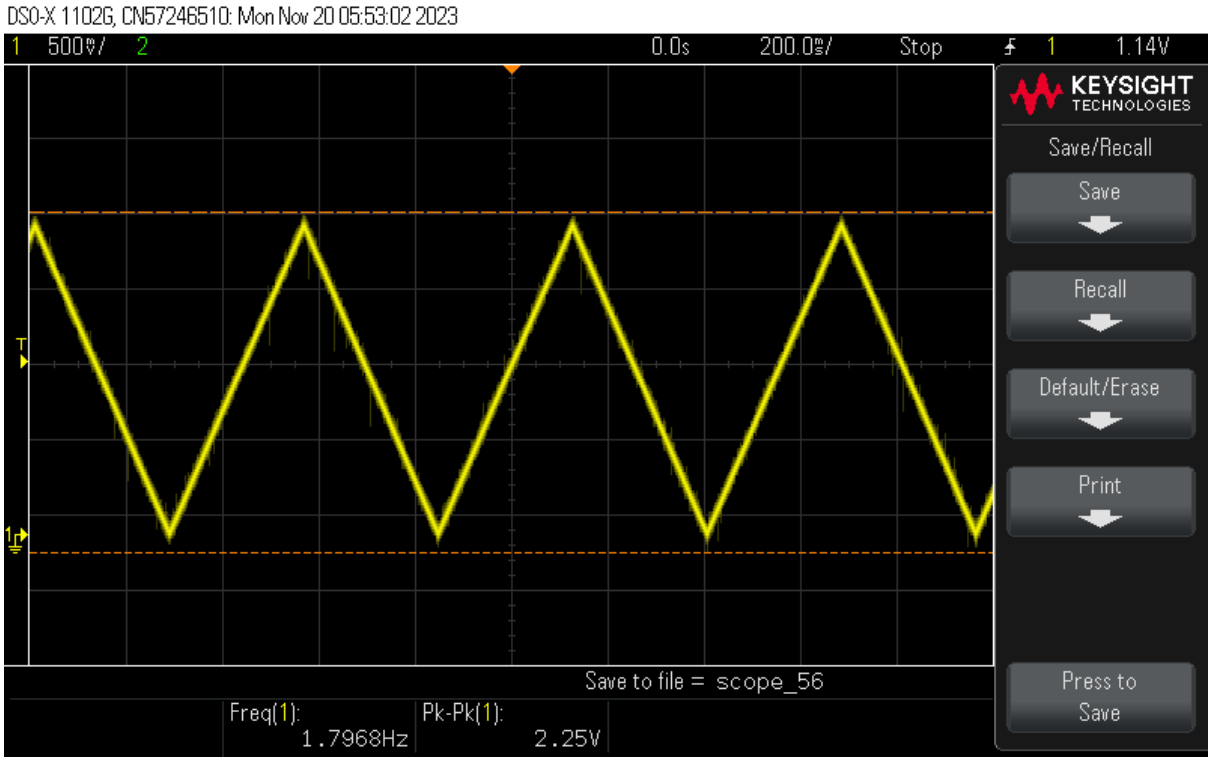
Oscilloscope picture of DAC used to generate a square wave.



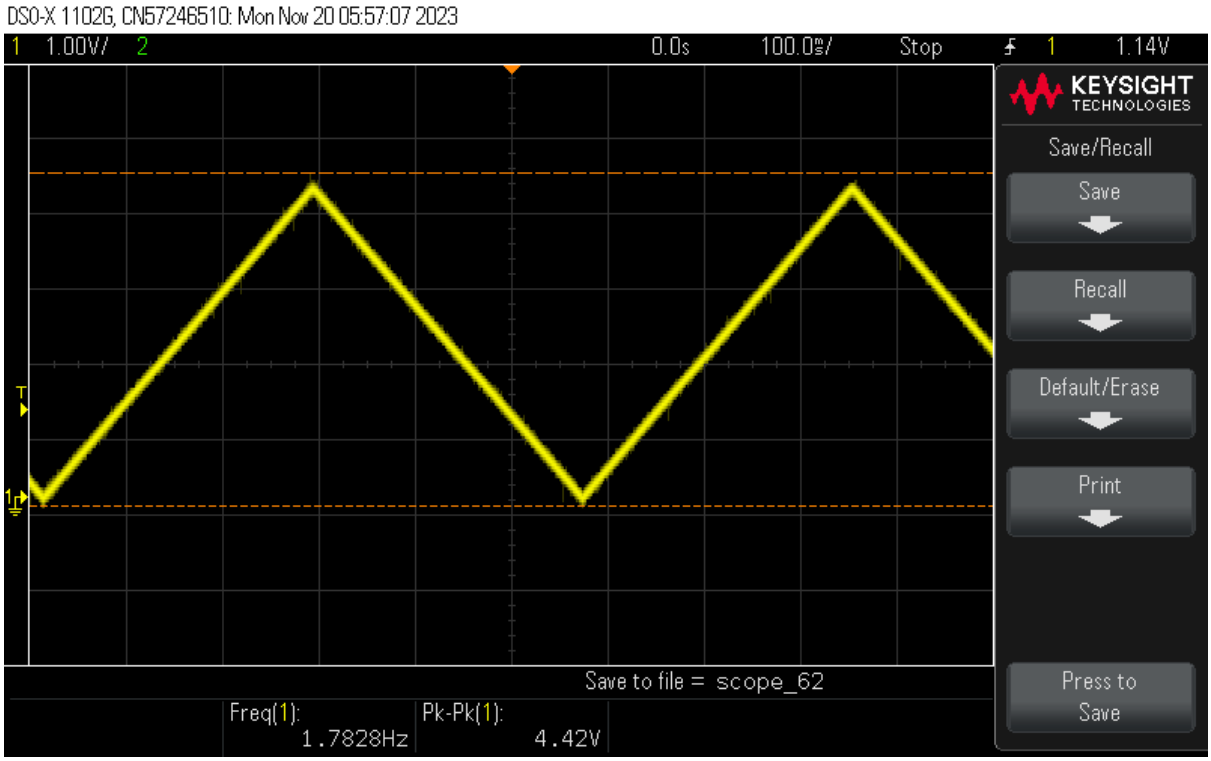
Oscilloscope picture of DAC used to generate a square wave with increased gain.



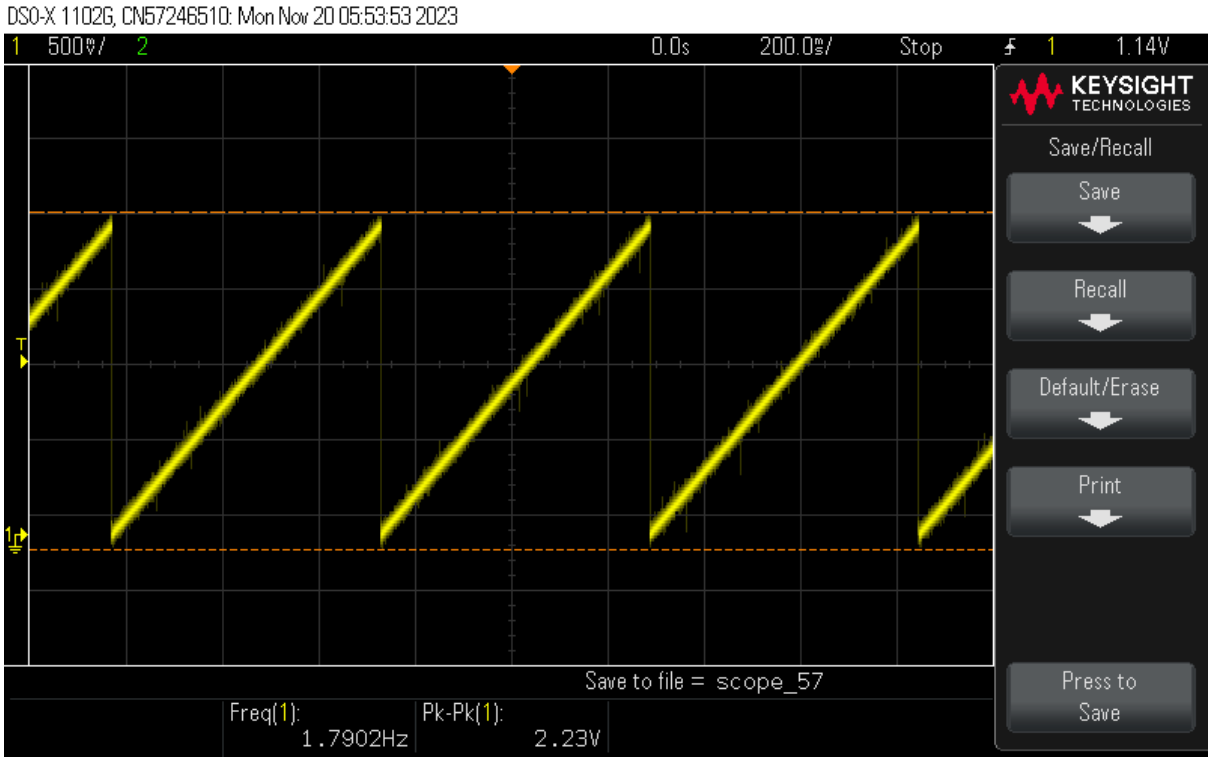
Oscilloscope picture of DAC used to generate the triangular wave.



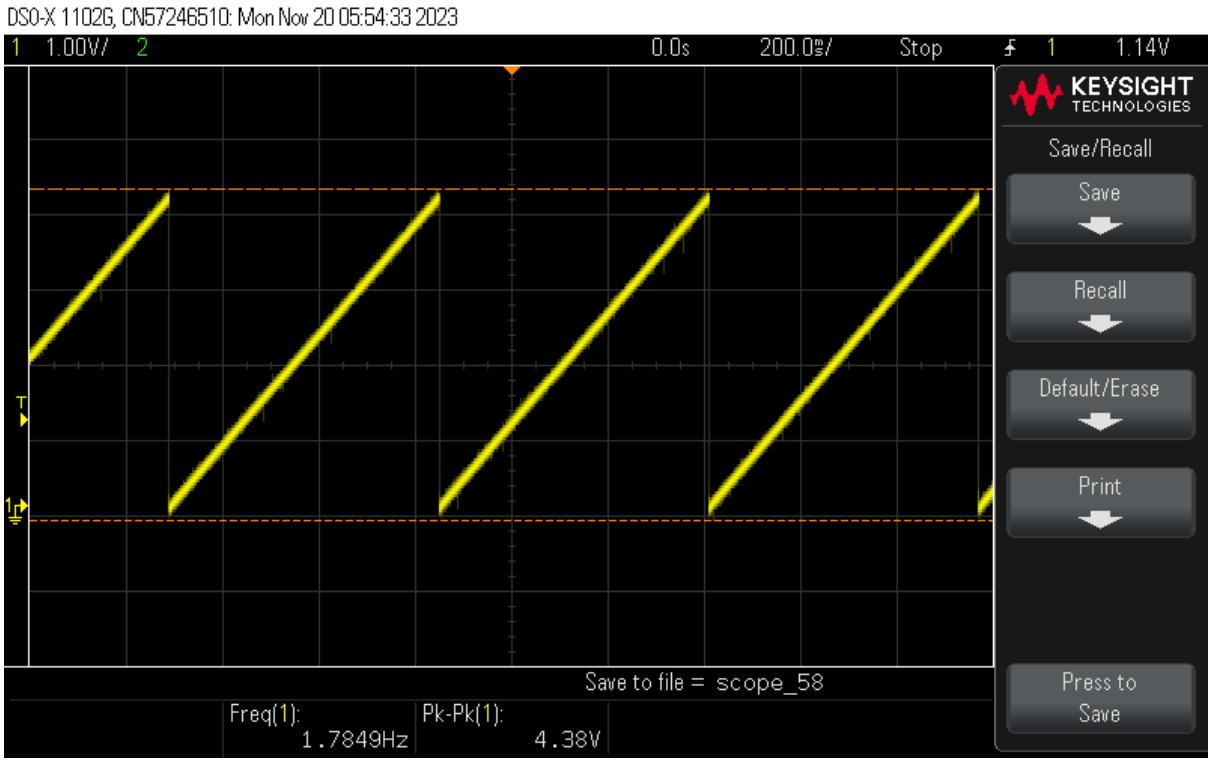
Oscilloscope picture of DAC used to generate the triangular wave with increased gain.



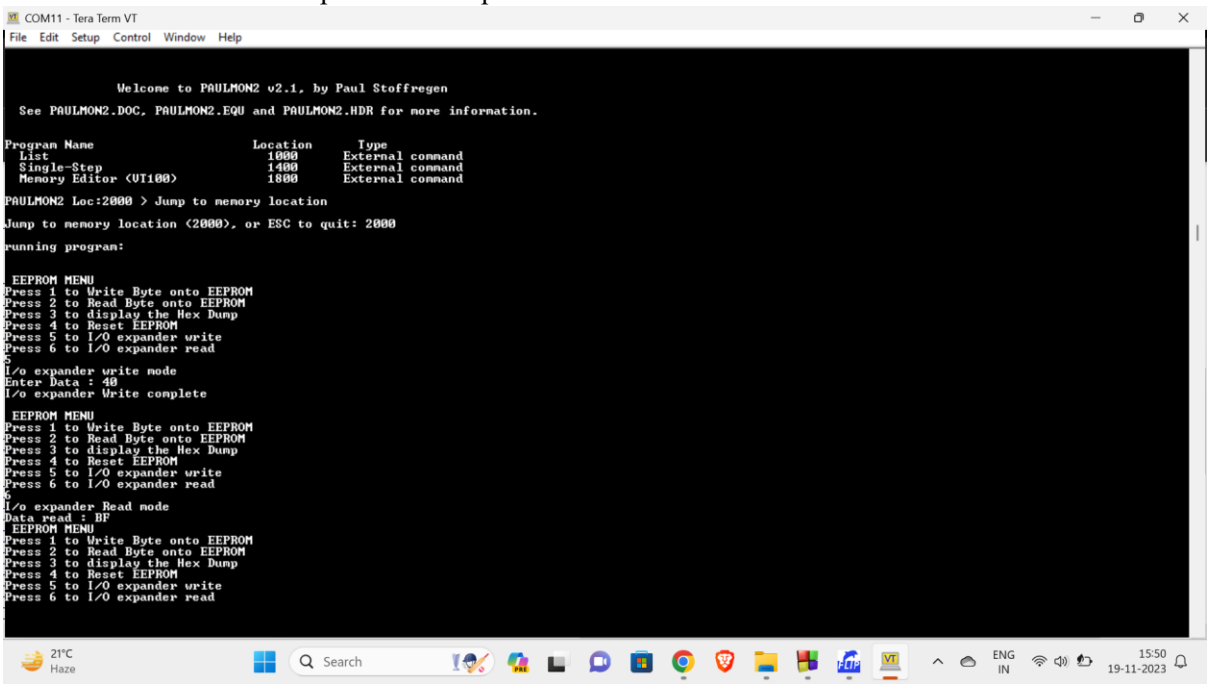
Oscilloscope picture of DAC used to generate the sawtooth wave.



Oscilloscope picture of DAC used to generate the sawtooth wave with increased gain.



3. Screenshot of Terminal output for I/O expander.



SIGNIFICANT LEARNINGS:

- I have learned how to write a driver and user interface for I2C protocol using EEPROM.
- I have learned to integrate LCD with 8051 and write the data on the LCD screen.
- I have learned to integrate a real-time clock with an LCD and control its operation like start, stop, and reset.
- I have learned to create a custom character and display it on the LCD.
- I have learned how to write a driver and user interface for SPI protocol using DAC.
- I have learned how to generate various waveforms like sinusoidal, square, triangular, and saw-tooth using DAC.
- I have learned how to implement an I/O expander using an I2C protocol that offers more digital I/O pins than what is available on the microcontroller.